IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Allen P. Chen et al.	Art Unit	: 2662
Serial No.	: 09/727,393	Examiner	: Gregory B. Sefcheck
Filed	: November 29, 2000	Conf. No.	: 4146
Title	: PACKET PROCESSING		

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

The required amount of $500 for the Appeal Brief fee is being paid concurrently herewith on the Electronic Filing System (EFS) by way of Deposit Account authorization. Please apply all charges or credits to Deposit Account No. 06-1050, referencing Attorney Docket No. 10559-385001.

(1) Real Party in Interest

The real party in interest in the above application is Intel Corporation, a corporation of California, having a place of business at 2200 Mission College Boulevard, Santa Clara, California 95052.

(2) Related Appeals and Interferences

The appellant is not aware of any appeals or interferences related to the above-identified patent application.

(3) Status of Claims

This is an appeal from the panel decision from a pre-appeal review[1] based on the rejections of claims 1-11 and 13-42 provided by the Examiner in a Final Office Action dated July 17, 2006. Claims 1-11 and 13-42 have been twice rejected and are presented for appeal.

(4) Status of Amendments

All amendments have been entered. Appellant previously filed a Notice of Appeal on November 9, 2006.

(5) Summary Of Claimed Subject Matter

Claim 1

Claim 1 is directed to a programmable intra-packet switching method. "FIG. 1 is a diagrammatic view of a programmable intra-packet switching process." [Appellant's specification page 2, lines 12-13].

Inventive features of claim 1 include determining which, if any, of a plurality of data ports connected to a network contains a data packet available for processing. "Intra-packet switching process 30 includes a port polling process 34 for polling, in a systematic fashion, the individual data ports 20_{1-N} connected to network 10 to determine if any of these ports 20_{1-N}

[1] In the decision from the pre-appeal review, the panel stated that the applicant should proceed to Board of Patent Appeals and Interferences because there was at least one actual issue for appeal.

contains a data packet for processing. For example, if data packets 28_{1-N} include two packets (namely packets XX and XY), where XX is a relatively large packet (1500 bytes) and XY is a relatively small packet (approximately 100 bytes), these two packets (XX and XY) would be stored in any two data ports (e.g., ports "A" and "B" respectively of data ports 20_{1-N}). As stated above, port polling process 34 systematically polls ports 20_{1-N} to determine if any port has a data packet available for processing. In this example, polling process 10 would poll port "A" of ports 20_{1-N} and determine that port "A" indeed has a data packet (XX) available for processing." [Appellant's specification page 4, line 21 – page 5, line 10].

Inventive features of claim 1 also include fragmenting a first portion of a first available data packet into at least one data cell having a defined size; wherein this fragmentation of the first data packet continues until a user-defined number of cells are generated. "Intra-packet switching process 30 includes a packet fragmentation process 36 which is responsive to port polling process 34 determining that port "A" of ports 20_{1-N} has a data packet (XX) available for processing. Packet fragmentation process 36 fragments packet (XX) stored on port "A" of ports 20_{1-N} into one or more data cells 38_{1-N}, where each cell has a predefined size 31... Packet fragmentation process 36 continues fragmenting data cell (XX) until a user-defined number of cells 33 are generated or, alternatively, the data packet (XX) is fully fragmented." [Appellant's specification page 5, line 11 – page 6, line 7].

Inventive features of claim 1 also include storing, in a memory, at least one data element concerning the first available data packet, wherein the at least one data element includes a data element indicative of the incomplete fragmentation status of said first available data packet; and the at least one data element enables fragmenting of a second portion of the first available data packet subsequent to fragmenting at least a portion of a second available data packet. "Whenever cell limit monitoring process 46 determines that the user defined number of cells have been generated and cell limit port switching process 50 determines that another data packet is available for processing, the processing (i.e. fragmentation) of the first data packet is paused in lieu of processing (i.e. fragmenting) the second data packet. Accordingly, in order to facilitate subsequent processing of the first packet's "chunk" reminder, information concerning the first data packet must be stored for later retrieval... packet information storage process 52 stores various data elements 54 concerning the data packet currently being processed... Data elements

54 can include information concerning: the overall length of the data packet currently being processed; the length of the packet remainder to be processed; the length of the portion of the packet that has already been processed; a packet truncation indicator showing that the packet has not been completely processed; or a Packet-Over-Sonet (POS) header for use in optical networks. Accordingly, by storing data elements 54 on non-volatile memory 32, subsequent fragmentation of the remainder of a data packet (not completely fragmented due to cell number limit 33) can be easily achieved." [Appellant's specification page 9, line 15 – page 10, line 18].

Inventive features of claim 1 also include subsequent to fragmenting the first portion of the first data packet and prior to fragmenting the second portion of the first available data packet, fragmenting at least a portion of the second available data packet on a different one of the plurality of data ports. "A cell monitoring process monitors 106 the number of data cells produced to determine if the user defined number of cells have been generated. If so, a cell limit port switching process initiates 108 the polling process to determine if any other port contains a data packet available for processing. If such a data packet is available for processing, a packet information storage process stores 110 at least one data element concerning the data packet currently being processed, where the data element allows for subsequent processing of the remainder of the data packet currently being processed. Further, if such a data packet is available for processing, a cell limit fragmentation switching process initiates 112 the packet fragmentation process to fragment the data packet on the other port...". [Appellant's specification page 12, lines 3-19].

Claim 9

Claim 9 is directed to a programmable intra-packet switching process. This feature has support as the analogous feature of claim 1.

Inventive features of claim 9 include a port polling process for determining which, if any, of a plurality of data ports connected to a network contains a data packet available for processing. This feature has support as the analogous feature of claim 1.

Inventive features of claim 9 include a packet fragmentation process, responsive to said port polling process determining that a first one of said ports contains a first data packet, for fragmenting said first data packet into at least one data cell having a defined size; wherein said

packet fragmentation process continues fragmenting said first data packet into said data cells until a user-defined number of cells are generated. This feature has support as the analogous feature of claim 1.

Inventive features of claim 9 include a packet information storage process for storing at least one data element concerning the first data packet, wherein said data element enables subsequent fragmenting of a second portion of the first data packet. This feature has support as the analogous feature of claim 1.

Inventive features of claim 9 include a second packet fragmentation process for fragmenting at least a portion of a second available data packet on a different one of the plurality of data ports subsequent to fragmenting the first portion of the first data packet and prior to fragmenting the second portion of the first available data packet. This feature has support as the analogous feature of claim 1.

Claim 21

Claim 21 is directed to a programmable packet fragmentation process. This feature has support as the analogous feature of claim 1.

Inventive features of claim 21 include a process for determining the availability of a data packet on a plurality of data ports connected to a synchronous optical network. "A typical example of network signal processor circuit 22 is an Intel IXF6402 Asynchronous Transfer Mode/Packet Over Sonet (ATM/POS) processor." [Appellant's specification page 4, lines 1-6]. This feature has additional support as the analogous feature of claim 1.

Inventive features of claim 21 include a packet fragmentation process, responsive to said process. This feature has support as the analogous feature of claim 1.

Inventive features of claim 21 include determining the availability of said data packet on one of said plurality of ports, for fragmenting a first portion of a first data packet into at least one Asynchronous Transfer Mode (ATM) cell, wherein said packet fragmentation process continues fragmenting said data packet into said data cells until a user-defined number of cells are generated. "While the size of these cells is user definable, a typical size is 53 bytes, which adheres to the Asynchronous Transfer Mode (ATM) guidelines." [Appellant's specification page 5, lines 20-22]. This feature has additional support as the analogous feature of claim 1.

Inventive features of claim 21 include a packet information storage process for storing at least one data element concerning the first data packet, wherein said data element enables fragmenting of a second portion of the first data packet subsequent to fragmenting at least a portion of a second available data packet. This feature has support as the analogous feature of claim 1.

Inventive features of claim 21 include, subsequent to fragmenting the first portion of the first data packet and prior to fragmenting the second portion of the first available data packet, fragmenting at least a portion of the second available data packet on a different one of the plurality of data ports. This feature has support as the analogous feature of claim 1.

Claim 24

Claim 24 is directed to a programmable intra-packet switching process. This feature has support as the analogous feature of claim 1.

Inventive features of claim 24 include a port polling process for determining which port, if any, of a plurality of data ports connected to a network contains a data packet available for processing. This feature has support as the analogous feature of claim 1.

Inventive features of claim 24 include a packet fragmentation process, responsive to said port polling process for determining that one of said ports contains a data packet. This feature has support as the analogous feature of claim 1.

Inventive features of claim 24 include a packet fragmentation process for fragmenting a first portion of a first data packet into at least one data cell; wherein said packet fragmentation process continues fragmenting said first data packet into said data cells until a port-switching event occurs. This feature has support as the analogous feature of claim 1.

Inventive features of claim 24 include a packet information storage process for storing at least one data element concerning the first data packet, wherein said data element enables fragmenting of a second portion of the first data packet subsequent to fragmenting at least a portion of a second available data packet. This feature has support as the analogous feature of claim 1.

Inventive features of claim 24 include, subsequent to a port-switching event packet and prior to fragmenting the second portion of the first available data packet, a process for

fragmenting at least a portion of the second available data packet on a different one of the plurality of data ports. This feature has support as the analogous feature of claim 1.

Claim 29

Claim 29 is directed to a computer program product residing on a computer readable medium having a plurality of instructions stored thereon. "Now referring to Fig. 3, there is shown a computer program product 200 which functions within a network signal processor circuit. Computer program product 200 resides on a computer readable medium 202 which has a plurality of instructions 203 stored thereon." [Appellant's specification page 13, lines 13-18].

Inventive features of claim 29 include instructions to determine which port, if any, of a plurality of data ports connected to a network contains a data packet available for processing. This feature has support as the analogous feature of claim 1.

Inventive features of claim 29 include instructions to fragment a first portion of a first available data packet into at least one data cell having a defined size; wherein this fragmentation continues until a user-defined number of cells are generated. This feature has support as the analogous feature of claim 1.

Inventive features of claim 29 include instructions to store at least one data element concerning the first available data packet, wherein the data element enables fragmenting of a second portion of the first available data packet subsequent to fragmenting at least a portion of a second available data packet. This feature has support as the analogous feature of claim 1.

Inventive features of claim 29 include instructions to fragment at least a portion of the second available data packet on a different one of the plurality of data ports, wherein this fragmentation occurs subsequent to the fragmentation of the first portion of the first data packet and prior to the fragmentation of the second portion of the first available data packet. This feature has support as the analogous feature of claim 1.

Claim 32

Claim 32 is directed to a processor and memory. "Now referring to Fig. 4, there is shown a processor 300 and memory 302 configured to poll 304, in a systematic fashion, a plurality of data ports connected to a network to determine which port, if any, contains a data packet

available for processing. Processor 300 and memory 302 fragment 306 the available data packet into at least one data cell having a defined size, where this fragmentation continues until a user-defined number of cells are generated." [Appellant's specification page 14, lines 6-13].

Inventive features of claim 32 include that the processor and memory are configured to determine which port, if any, of a plurality of data ports connected to a network contains a data packet available for processing. This feature has support as the analogous feature of claim 1.

Inventive features of claim 32 include that the processor and memory are configured to fragment a first portion of a first available data packet into at least one data cell having a defined size; wherein this fragmentation continues until a user-defined number of cells are generated. This feature has support as the analogous feature of claim 1.

Inventive features of claim 32 include that the processor and memory are configured to store at least one data element concerning the first available data packet, wherein the data element enables fragmenting of a second portion of the first available data packet subsequent to fragmenting at least a portion of a second available data packet. This feature has support as the analogous feature of claim 1.

Inventive features of claim 32 include that the processor and memory are configured to fragment at least a portion of the second available data packet on a different one of the plurality of data ports, wherein this fragmentation occurs subsequent to the fragmentation of the first portion of the first data packet and prior to the fragmentation of the second portion of the first available data packet. This feature has support as the analogous feature of claim 1.

Claim 42

Claim 42 is directed to a programmable intra-packet switching process. This feature has support as the analogous feature of claim 1.

Inventive features of claim 42 include determining which, if any, of a plurality of data ports connected to a network contains a data packet available for processing. This feature has support as the analogous feature of claim 1.

Inventive features of claim 42 include fragmenting at least a first portion of a first available data packet into at least one data cell having a defined size. This feature has support as the analogous feature of claim 1.

Inventive features of claim 42 include monitoring the number of data cells produced to determine if a user defined number of cells have been generated. "A cell limit monitoring process 46 monitors the number of cells 381-N produced by packet fragmentation process 36 to determine if the user-defined number of cells 33 has been produced." [Appellant's specification page 6, lines 16-19].

Inventive features of claim 42 include re-determining which, if any, of the plurality of data ports contains a data packet available for processing, if it is determined that the user defined number of cells have been generated, to determine if any other port contains a data packet available for processing. "Cell limit monitoring process 46 includes a cell limit port switching process 48 which is responsive to cell limit monitoring process 46 determining that the user defined number of cells 33 have been generated. If this occurs, cell limit port switching process 48 initiates port polling process 34 to determine if any other port contains a data packet which is available for processing." [Appellant's specification page 7, lines 11-17].

Inventive features of claim 42 include storing at least one data element concerning the data packet currently being processed if it is determined that another port contains a data packet available for processing where the at least one data element includes a first data element indicative of the incomplete fragmentation status of said first available data packet and a second data element selected from the group consisting of a data element indicative of the length of the portion of said data packet not fragmented and a data element indicative of the length of the portion of said data packet previously fragmented and the at least one data element enables fragmenting of a second portion of the first available data packet subsequent to fragmenting at least a portion of a second available data packet. "Whenever cell limit monitoring process 46 determines that the user defined number of cells have been generated and cell limit port switching process 50 determines that another data packet is available for processing, the processing (i.e. fragmentation) of the first data packet is paused in lieu of processing (i.e. fragmenting) the second data packet. Accordingly, in order to facilitate subsequent processing of the first packet's "chunk" reminder, information concerning the first data packet must be stored for later retrieval... packet information storage process 52 stores various data elements 54 concerning the data packet currently being processed... Data elements 54 can include information concerning: the overall length of the data packet currently being processed; the

length of the packet remainder to be processed; the length of the portion of the packet that has already been processed; a packet truncation indicator showing that the packet has not been completely processed; or a Packet-Over-Sonet (POS) header for use in optical networks. Accordingly, by storing data elements 54 on non-volatile memory 32, subsequent fragmentation of the remainder of a data packet (not completely fragmented due to cell number limit 33) can be easily achieved." [Appellant's specification page 9, line 15 – page 10, line 18].

Inventive features of claim 42 include, subsequent to fragmenting the first portion of the first data packet and prior to fragmenting the second portion of the first available data packet, fragmenting at least a portion of the second available data packet on a different one of the plurality of data ports. "Further, if such a data packet is available for processing, a cell limit fragmentation switching process initiates 112 the packet fragmentation process to fragment the data packet on the other port..." [Appellant's specification page 12, lines 12-19].

Inventive features of claim 42 include, subsequent to fragmenting the first portion of the second data packet, fragmenting a second portion of the first available data packet. "If such a data packet is available for processing, a packet information storage process stores 110 at least one data element concerning the data packet currently being processed, where the data element allows for subsequent processing of the remainder of the data packet currently being processed." [Appellant's specification page 12, lines 7-12].

(6) Grounds of Rejection to be Reviewed on Appeal

1. Claims 1-4, 6-7, 9-11, 13, 15-16, 18-19, 24, 26-33, and 35-42 were rejected under 35 U.S.C. §103(a) as being unpatentable over Cam in view of Bucholz.

2. Claims 5, 8, 14, and 17 were rejected under 35 U.S.C. §103(a) as being unpatentable over Cam in view of Bucholz in further view of Colmant.

3. Claims 20-23 and 43 were rejected under 35 U.S.C. §103(a) as being unpatentable over Cam in view of Bucholz in further view of Jha.

4. Claim 25 was rejected under 35 U.S.C. §103(a) as being unpatentable over Cam in view of Bucholz in further view of Muller.

(7) Argument

1. Cam in view of Bucholz fails to render obvious claims 1-4, 6-7, 9-11, 13, 15-16, 18-19, 24, 26-33, and 35-42.

Claims 1, 3, 6-7, 9, 11, 13, 15-16, 18-19, 24, 26, 28-33, and 35-42.

For the purposes of this appeal only, claims 1, 3, 6-7, 9, 11, 13, 15-16, 18-19, 24, 26, 28-33, and 35-42 stand or fall together. Claim 1 is representative of this group of claims.

Cam, alone or in combination with Bucholz, fails to disclose or suggest "fragmenting a first portion of a first available data packet into at least one data cell having a defined size; wherein this fragmentation of the first data packet continues until a user-defined number of cells are generated," as recited in the Appellant's claim 1.

At the outset, it is Appellant's contention that Cam does not relate to packet fragmentation, let alone the feature of packet fragmentation recited in Appellant's claim 1. Rather, Cam teaches a method for "coordinating the transfer of data" based on a process that includes polling, selection, and data transfer.[2]

In Cam's data transfer process, a device polls multiple ports to determine which, if any, ports have data available for transfer.[3] After determining which ports have data available for transfer, the device selects a particular port and transfers data from the selected port.[4] As noted

[2] Cam, col. 2, lines 55-60.
[3] Id., col. 2, line 60 – col. 3, line 10.
[4] Id., col. 3, lines 30-45.

by the Examiner, Cam's process limits a maximum block size of data that may be transferred.[5] For example, Cam states:[6]

> For both the transmit and receive interfaces, the maximum block size that may be transferred depends on the application. For example, the maximum block size for ATM cells may be 52-bytes (excluding the header error control byte) while the maximum block size for packet fragments may be 256 bytes. The maximum block size is fixed at start-up, either inherently in the Link Layer and PHY-devices, or by programming through an external management interface.

Thus, Cam limits the size of a data transfer not packet fragmentation. As such, Cam fails to disclose or suggest a method in which "fragmentation of the first data packet continues until a user-defined number of cells are generated." Bucholz does not add any teaching that would remedy this deficiency of Cam.

Additionally, Cam, alone or in combination with Bucholz, fails to disclose or suggest a data element that "enables fragmenting of a second portion of the first available data packet subsequent to fragmenting at least a portion of a second available data packet" as recited in the Appellant's claim 1.

As the examiner acknowledges,[7] Cam does not disclose or suggest storing a data element that enables fragmenting of a second portion of a first available data packet subsequent to fragmenting at least a portion of a second available data packet.

Bucholz discloses a packet acknowledgement system to assure delivery of all fragments of a fragmented data packet.[8] The reassembly header to which the examiner refers simply provides information for re-assembling packet fragments. Therefore, unlike the appellant's data element, Bucholz's reassembly header 430 does not provide information to enable fragmenting

---

[5] Id., col. 3, lines 2-4.
[6] Id., col. 3, line 2-10 (emphasis added).
[7] Final Office Action mailed July 17, 2006, page 4.
[8] Bucholz, abstract.

of second portion of a first available data packet subsequent to fragmenting at least a portion of a second available data packet.

More particularly, Bucholz fragments an entire packet and sends the fragmented packet in consecutive TDMA frames.[9] Upon receipt, the reassembly header 430 is used to re-assemble the fragmented packet from the multiple packet fragments. The examiner contends that: "... Bucholz shows that a reassembly header (stored data element) is stored in the fragmented packet ..."

Appellant disagrees that Bucholz's reassembly header corresponds to the data element in Appellants step of "storing, in a memory, at least one data element concerning the first available data packet." Because Bucholz fragments the entire packet before Bucholz fragments another packet, and sends out the TDMA frames consecutively, the information included in Bucholz's reassembly header 430 is used reassemble the fragments. The reassembly header 430 is not the "data element concerning the first available data packet" since it would not permit Bucholz to fragment a second portion of the first available data packet subsequent to fragmenting at least a portion of a second available data packet, as recited in the appellant's claim 1. Moreover, since the packet in Bucholz is fragmented consecutively, Bucholz would have no need for a data element that enables fragmenting of a second portion of a first available data packet subsequent to fragmenting at least a portion of a second available data packet. Therefore, Bucholz's reassembly header cannot be an example of the Appellant's "data element."

Claims 9, 21, and 24 all recite "a packet information storage process for storing at least one data element concerning the first data packet, wherein said data element enables fragmenting

[9] Id., col. 8, lines 9-13.

of a second portion of the first data packet subsequent to fragmenting at least a portion of a second available data packet." Accordingly, Cam, alone or in combination with Bucholz, fails to render obvious claims 9, 21, and 24 for at least the same reasons noted above with respect to claim 1.

Claims 29 and 32 recite "storing at least one data element concerning the first available data packet, wherein the data element enables fragmenting of a second portion of the first available data packet subsequent to fragmenting at least a portion of a second available data packet." Accordingly, Cam, alone or in combination with Bucholz, fails to render obvious claims 29 and 32 for at least the same reasons noted above with respect to claim 1.

Claims 2, 10, and 27

For the purposes of this appeal only, claims 2, 10 and 27 stand or fall together. Claim 2 is representative of this group of claims.

Claim 2 depends on independent claim 1 and recites the additional limitation of "monitoring the number of data cells produced to determine if the user defined number of cells have been generated." The examiner states that Cam shows monitoring the number of cells produced to determine if [a] user defined number are generated.[10] However, as described above, Cam teaches a method for "coordinating the transfer of data" based on a process that includes polling, selection, and data transfer.[11] As such, Cam does not even relate to packet fragmentation let alone disclose the method of packet fragmentation that includes monitoring the number of data cells produced as covered by the Appellant's claims.

---

[10] Office Action, page 3
[11] Cam, col. 2, lines 55-60.

Accordingly, Cam whether taken alone or in combination with Bucholz fails to render obvious claim 2 since at least these features of claim 2 are not described in the references.

Claim 4

Claim 4 depends on claim 3 and recites the additional limitation of "storing at least one data element concerning the data packet currently being processed if it is determined that another port contains a data packet available for processing, wherein this data element allows for subsequent processing of the remainder of the data packet currently being processed."

As the examiner acknowledges, Cam does not disclose or suggest storing such a data element.[12]

Bucholz's reassembly header 430 includes a logical unit identification (LUID) 610, a packet identification (ID) field 620, a sequence number field 630, total fragment field 640, a fragment number field 650, a total packet length field 660, and a protocol field 670.[13] None of these fields describe or would have made obvious a data element indicative of an incomplete fragmentation status. Even when viewed in combination the fields in Bucholz's reassembly header, the fields in Bucholz's reassembly header simply provide information for re-assembling packet fragments and are not used to enable "subsequent processing of the remainder of the data packet currently being processed." Each of the fields of Bucholz's reassembly header 430 are addressed below.

Logical unit identification (LUID) 610 – According to Bucholz the logical unit identification 610 "defines the logical unit identification of the originating device."[14] Identification of the originating device does not enable "subsequent processing of the remainder of the data packet currently being processed." Therefore, Bucholz's LUID 610 cannot be the appellant's data element.

Packet identification (ID) field 620 and sequence number field 630 – According to Bucholz the packet identification (ID) field 620 and the sequence number field 630 "in

[12] Office Action, page 4.
[13] Bucholz , col.6, line 67 – col. 7, line 4.
[14] Id., col. 7, lines 5-6

combination, are used to provide a unique ID for each data packet."[15] Thus, fields 620 and 630 are used merely to identify the packet for reassembly and do not enable "subsequent processing of the remainder of the data packet currently being processed." Therefore, Bucholz's fields 620 and 630 cannot be the appellant's data element.

Total fragment field 640 - According to Bucholz the total fragment field 640 "defines the total number of fragments comprising the data packet in question."[16] Identification of the total number of fragments does not enable "subsequent processing of the remainder of the data packet currently being processed." Therefore, Bucholz's total fragment field 640 cannot be the appellant's data element.

Fragment number field 650 – According to Bucholz the fragment number field 650 "defines which of the fragments is being received."[17] The identification of a fragment does not enable "subsequent processing of the remainder of the data packet currently being processed" and therefore, cannot be the appellant's data element.

Total packet length field 660 - According to Bucholz the total packet length field 660 "defines the length in bytes of the data packet as reassembled."[18] The total length of the packet does not enable "subsequent processing of the remainder of the data packet currently being processed." Therefore, field 660 cannot be the appellant's data element.

Protocol field 670 - According to Bucholz the protocol field 670 is employed to "assure the proper receipt of each fragment."[19] Since the Protocol field 670 is associated with the receipt status it does not enable "subsequent processing of the remainder of the data packet currently being processed." Therefore, protocol field 670 cannot be the appellant's data element.

As shown above, none of the fields in Bucholz's reassembly header, to which the examiner refers, allow for subsequent processing of the remainder of the data packet currently being processed and Appellant contends that this feature is neither described nor suggested by any combination of Cam with Bucholz.

---

[15] Id., col. 7, lines 7-9
[16] Id., col. 7, lines 15-16
[17] Id., col. 7, lines 17-18
[18] Id., col. 7, lines 19-20
[19] Id., col. 7, lines 20-23

2) Cam in view of Bucholz in further view of Colmant fails to render obvious claims 5, 8, 14, and 17.

Claims 5, 8, 14, and 17.

For the purposes of this appeal only, claims 5, 8, 14, and 17 stand or fall together. Claim 5 is representative of this group of claims.

Claim 5 depends on independent claim 1 and recites the additional limitation that monitoring the number of data cells produced includes "initiating the fragmentation process, if it is determined that another port contains a data packet for processing, to fragment the data packet on the other port into at least one data cell having a defined size; wherein the packet fragmentation process continues fragmenting the data packet on the other port into data cells until the user-defined number of cells are generated."[20] Claim 5 is not described or suggested in Cam whether taken alone or in combination with Bucholz and Colmant.

Colmant relates to a switching device for fixed-size packets of data such as ATM packets.[21] Colmant does not relate to packet fragmentation. While Colmant does use the word "fragment," Colmant uses this word to refer to storing a lengthy packet in multiple memory locations.[22] The packet is later sent as a single packet by reading each of the memory locations in sequence.[23] As such, Colmant does not disclose or suggest "initiating the fragmentation

---

[20] It appears that the examiner incorrectly interpreted claim 5 to require simultaneous fragmentation. For example, the examiner states "Cam does not explicitly show initiating fragmentation on a data packet from another port while the fragmentation of the data packet on the first port continues until the user-defined cells are generated." The examiner also states that "[b]y fragmenting received packets in parallel, processing delay required to fragment a packet prior to transmission could be eliminated." In contrast to the examiner's interpretation, claim 5 requires that the fragmentation on the other port (not the first port) continues until the user-defined number of cells are generated.

[21] Colmant, col. 1, lines 8-9.

[22] See, e.g., Colmant, col. 10, lines 42-48.

[23] See, e.g., Colmant, col. 10, lines 53-55.

process... to fragment the data packet on the other port... wherein the packet fragmentation process continues fragmenting the data packet on the other port into data cells until the user-defined number of cells are generated" as recited in appellant's claim 5.

The examiner acknowledges that Cam and Bucholz fail to disclose this feature, so accordingly, Cam whether taken alone or in combination with Bucholz and Colmant fails to render obvious claim 5 since at least these features of claim 5 are not described in any combination of the cited references.

Claims 8, 14, and 17 roughly correspond to claim 5. Accordingly, Cam whether taken alone or in combination with Bucholz and Colmant fails to render obvious claims 8, 14, and 17 for at least the same reasons noted above with respect to claim 5.

3) Cam in view of Bucholz in further view of Jha fails to render obvious claims 20-23 and 43.

For the purposes of this appeal only, claims 20-23 and 43 stand or fall together. Claim 20 is representative of this group of claims.

Claim 20 depends on independent claim 9 and recites the additional limitation that the "at least one data cell having a defined size is a 53-byte Asynchronous Transfer Mode (ATM) cell."

Appellant contends that this rejection is an exercise in improper hindsight reconstruction, using appellant's claim as a template to reconstruct the invention by picking and choosing isolated disclosures from the prior art. This is impermissible under the law.[24] The present rejection fits the court's description of what is prohibited in formulation of a rejection under §103. The examiner has merely listed certain components of applicant's invention and then

[24] In re Fritch, 972 F.2d 1260, 1266, 23 USPQ2d 1780, 1784 (Fed. Cir. 1992)

located isolated disclosures of those components. The law requires more than that. The examiner must show where the prior art provides a motivation to combine the references he/she has combined in the obviousness rejection. Absent a motivation to combine, obviousness has not been demonstrated.[25]

In the rejection the examiner simply states that "it would have been obvious to one of ordinary skill in the art at the time of the invention to implement the method of Cam through an ATM/POS processor such that the plurality of ports are connected to a SONET network and the fragmentation process produces ATM cells, as shown by Jha."[26] The examiner has not provided any indication of where such a motivation is found in Cam. Simply showing that the ATM protocol existed at the time of the invention is not enough to satisfy the requirement of providing a motivation to modify Cam to use the protocol.

Accordingly, Cam whether taken alone or in combination with Bucholz and Jha fails to render obvious claim 20 since at least these features of claim 20 are not described in the references.

4) Cam in view of Bucholz in further view of Muller fails to render obvious claim 25.

Claim 25 depends on independent claim 24 and recites the additional limitation that a "port-switching event is an unbalanced port-loading condition." Claim 25 is not described or suggested in Cam whether taken alone or in combination with Bucholz and Muller.

---

[25] Northern Telecom, Inc. v. Datapoint Corp., 908 F.2d 931, 934, 15 USPQ2d 1321, 1323 (Fed. Cir. 1990)
[26] Office Action, page 14.

The examiner states that Cam does not disclose an unbalanced port-loading condition as a port switching event.

As with several of the other references cited by the examiner, Muller also does not relate to packet fragmentation. As noted by the examiner, Muller relates to a switch configuration where messages are directed between different nodes to balance loading of the nodes.[27] The appellant's claim requires that a "port-switching event" is an unbalanced port-loading condition where packet fragmentation continues until such a port-switching event occurs. Since Muller simply directs packets based on load conditions and does not relate to packet fragmentation at all, Muller's load balancing of messages through the nodes cannot be a "port-switching event" as in the appellant's claim 25.

Accordingly, Cam whether taken alone or in combination with Bucholz and Muller fails to render obvious claim 25 since at least these features of claim 25 are not described in the references.

Conclusion

Appellant submits, therefore, that Claims 1-12, 14-20, and 28-34 are allowable over the cited art. Therefore, the Examiner erred in rejecting Appellant's claims and should be reversed.

[27] Muller, col. 26, line 61 to col. 27, line 5.

Respectfully submitted,

Date: 2/28/07

Tonya S. Drake
Reg. No. 57,861

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21571205.doc

Appendix of Claims

1. A programmable intra-packet switching method comprising:

determining which, if any, of a plurality of data ports connected to a network, contains a data packet available for processing;

fragmenting a first portion of a first available data packet into at least one data cell having a defined size; wherein this fragmentation of the first data packet continues until a user-defined number of cells are generated;

storing, in a memory, at least one data element concerning the first available data packet, wherein:

the at least one data element includes a data element indicative of the incomplete fragmentation status of said first available data packet; and

the at least one data element enables fragmenting of a second portion of the first available data packet subsequent to fragmenting at least a portion of a second available data packet; and

subsequent to fragmenting the first portion of the first data packet and prior to fragmenting the second portion of the first available data packet, fragmenting at least a portion of the second available data packet on a different one of the plurality of data ports.

2. The programmable intra-packet switching method of claim 1 further comprising:

monitoring the number of data cells produced to determine if the user defined number of cells have been generated.

3. The programmable intra-packet switching method of claim 2 wherein monitoring the number of data cells produced includes:

re-determining which, if any, of the plurality of data ports contains a data packet available for processing, if it is determined that the user defined number of cells have been generated, to determine if any other port contains a data packet available for processing.

4. The programmable intra-packet switching method of claim 3 wherein monitoring the number of data cells produced includes:

storing at least one data element concerning the data packet currently being processed if it is determined that another port contains a data packet available for processing, wherein this data element allows for subsequent processing of the remainder of the data packet currently being processed.

5. The programmable intra-packet switching method of claim 4 wherein monitoring the number of data cells produced includes:

initiating the fragmentation process, if it is determined that another port contains a data packet for processing, to fragment the data packet on the other port into at least one data cell having a defined size; wherein the packet fragmentation process continues fragmenting the data packet on the other port into data cells until the user-defined number of cells are generated.

6. The programmable intra-packet switching method of claim 1 further comprising:

determining if the data packet has been fully fragmented into at least one data cell.

7. The programmable intra-packet switching method of claim 6 wherein determining if the data packet has been fully fragmented includes:

re-determining which, if any, of the plurality of data ports contains a data packet available for processing, if it is determined that the data packet has been fully fragmented into at least one data cell, to determine if any other port contains a data packet available for processing.

8. The programmable intra-packet switching method of claim 7 wherein determining if the data packet has been fully fragmented includes:

initiating the fragmentation process, if it is determined that another port contains a data packet for processing, to fragment the data packet on the other port into at least one

data cell having a defined size; wherein the packet fragmentation process continues fragmenting the data packet on the other port into data cells until the user-defined number of cells are generated.

9. A programmable intra-packet switching process comprising:

a port polling process for determining which, if any, of a plurality of data ports connected to a network contains a data packet available for processing;

a packet fragmentation process, responsive to said port polling process determining that a first one of said ports contains a first data packet, for fragmenting said first data packet into at least one data cell having a defined size; wherein said packet fragmentation process continues fragmenting said first data packet into said data cells until a user-defined number of cells are generated;

a packet information storage process for storing at least one data element concerning the first data packet, wherein said data element enables subsequent fragmenting of a second portion of the first data packet; and

a second packet fragmentation process for fragmenting at least a portion of a second available data packet on a different one of the plurality of data ports subsequent to fragmenting the first portion of the first data packet and prior to fragmenting the second portion of the first available data packet.

10. The programmable intra-packet switching process of claim 9 further comprising:

a cell limit monitoring process for monitoring the number of data cells produced by said packet fragmentation process to determine if said user defined number of cells have been generated.

11. The programmable intra-packet switching process of claim 10 wherein said cell limit monitoring process includes:

a cell limit port switching process, responsive to said cell limit monitoring process determining that said user defined number of cells have been generated, for initiating said

polling process to determine if any other port contains a data packet available for processing.

Claim 12 is canceled.

13. The programmable intra-packet switching process of claim 9 wherein said at least one data element includes:

a data packet remainder length indicator, indicative of the length of the portion of said data packet not fragmented; and

a packet truncation indicator, indicative of the incomplete fragmentation status of said data packet.

14. The programmable intra-packet switching process of claim 11 wherein said cell limit monitoring process includes:

a cell limit fragmentation switching process, responsive to said cell limit port switching process determining that another port contains a data packet for processing, for initiating said packet fragmentation process to fragment said data packet on said other port into at least one data cell having a defined size; wherein said packet fragmentation process continues fragmenting said data packet on said other port into said data cells until said user-defined number of cells are generated.

15. The programmable intra-packet switching process of claim 9 further comprising:

a packet completion monitoring process for monitoring the status of said packet fragmentation process to determine if said data packet has been fully fragmented into said at least one data cell.

16. The programmable intra-packet switching process of claim 15 wherein said packet completion monitoring process includes:

a packet completion port switching process, responsive to said packet completion monitoring process determining that said data packet has been fully fragmented into said

at least one data cell, for initiating said polling process to determine if any other port contains a data packet available for processing.

17. The programmable intra-packet switching process of claim 16 wherein said packet completion monitoring process includes:

a packet completion fragmentation switching process, responsive to said packet completion port switching process determining that another port contains a data packet for processing, for initiating said packet fragmentation process to fragment said data packet on said other port into at least one data cell having a defined size; wherein said packet fragmentation process continues fragmenting said data packet on said other port into said data cells until said user-defined number of cells are generated.

18. The programmable intra-packet switching process of claim 9 further comprising:

a user interface for allowing a user to specify at least one user-defined parameter utilized by said packet fragmentation process.

19. The programmable intra-packet switching process of claim 18 wherein said at least one user-defined parameter includes:

said user-defined number of cells to be generated by said packet fragmentation process; and

said defined size of said at least one data cell.

20. The programmable intra-packet switching process of claim 9 wherein said at least one data cell having a defined size is a 53-byte Asynchronous Transfer Mode (ATM) cell.

21. A programmable packet fragmentation process comprising:

a process for determining the availability of a data packet on a plurality of data ports connected to a synchronous optical network;

a packet fragmentation process, responsive to said process, said packet fragmentation process including:

determining the availability of said data packet on one of said plurality of ports, for fragmenting a first portion of a first data packet into at least one Asynchronous Transfer Mode (ATM) cell, wherein said packet fragmentation process continues fragmenting said data packet into said data cells until a user-defined number of cells are generated;

a packet information storage process for storing at least one data element concerning the first data packet, wherein said data element enables fragmenting of a second portion of the first data packet subsequent to fragmenting at least a portion of a second available data packet; and

subsequent to fragmenting the first portion of the first data packet and prior to fragmenting the second portion of the first available data packet, fragmenting at least a portion of the second available data packet on a different one of the plurality of data ports.

22. The programmable intra-packet switching process of claim 21 further comprising:

a cell limit monitoring process for monitoring the number of data cells produced by said packet fragmentation process to determine if said user defined number of cells have been generated.

23. The programmable intra-packet switching process of claim 21 further comprising:

a packet completion monitoring process for monitoring the status of said packet fragmentation process to determine if said data packet has been fully fragmented into said at least one data cell.

24. A programmable intra-packet switching process comprising:

a port polling process for determining which port, if any, of a plurality of data ports connected to a network contains a data packet available for processing;

a packet fragmentation process, responsive to said port polling process:

for determining that one of said ports contains a data packet,

for fragmenting a first portion of a first data packet into at least one data cell; wherein said packet fragmentation process continues fragmenting said first data packet into said data cells until a port-switching event occurs;

a packet information storage process for storing at least one data element concerning the first data packet, wherein said data element enables fragmenting of a second portion of the first data packet subsequent to fragmenting at least a portion of a second available data packet; and

subsequent to a port-switching event packet and prior to fragmenting the second portion of the first available data packet, for fragmenting at least a portion of the second available data packet on a different one of the plurality of data ports.

25. The programmable intra-packet switching process of claim 24 wherein said port-switching event is an unbalanced port-loading condition.

26. The programmable intra-packet switching process of claim 24 wherein said port-switching event is the generation of a user-defined number of cells.

27. The programmable intra-packet switching process of claim 26 further comprising:

a cell limit monitoring process for monitoring the number of data cells produced by said packet fragmentation process to determine if said user defined number of cells have been generated.

28. The programmable intra-packet switching process of claim 24 further comprising:

a packet completion monitoring process for monitoring the status of said packet fragmentation process to determine if said data packet has been fully fragmented into said at least one data cell.

29. A computer program product residing on a computer readable medium having a plurality of instructions stored thereon which, when executed by the processor, cause that processor to:

determine which port, if any, of a plurality of data ports connected to a network contains a data packet available for processing;

fragment a first portion of a first available data packet into at least one data cell having a defined size; wherein this fragmentation continues until a user-defined number of cells are generated;

storing at least one data element concerning the first available data packet, wherein the data element enables fragmenting of a second portion of the first available data packet subsequent to fragmenting at least a portion of a second available data packet; and

fragment at least a portion of the second available data packet on a different one of the plurality of data ports, wherein this fragmentation occurs subsequent to the fragmentation of the first portion of the first data packet and prior to the fragmentation of the second portion of the first available data packet.

30. The computer program product of claim 29 wherein said computer readable medium is a read-only memory.

31. The computer program product of claim 29 wherein said computer readable medium is a random access memory.

32. A processor and memory configured to:

determine which port, if any, of a plurality of data ports connected to a network contains a data packet available for processing;

fragment a first portion of a first available data packet into at least one data cell having a defined size; wherein this fragmentation continues until a user-defined number of cells are generated;

storing at least one data element concerning the first available data packet, wherein the data element enables fragmenting of a second portion of the first available data packet subsequent to fragmenting at least a portion of a second available data packet; and

fragment at least a portion of the second available data packet on a different one of the plurality of data ports, wherein this fragmentation occurs subsequent to the fragmentation of the first portion of the first data packet and prior to the fragmentation of the second portion of the first available data packet.

33. The processor and memory of claim 32 wherein said processor and memory are incorporated into a single board computer.

34. The processor and memory of claim 32 wherein said processor and memory are incorporated into an Asynchronous Transfer Mode / Packet Over Sonet (ATM/POS) processor.

35. The method of claim 1, wherein the at least one data element includes:

a data packet remainder length indicator, indicative of the length of the portion of said data packet not fragmented.

36. The method of claim 1, wherein the at least one data element includes:

an indicator indicative of the length of the portion of said data packet previously fragmented; and

an indicator indicative of a total length of the data packet.

37. The method of claim 1, wherein the at least one data element includes:

a data packet remainder length indicator, indicative of the length of the portion of said data packet not fragmented; and

an indicator indicative of a total length of the data packet.

38. The method of claim 1, further comprising, subsequent to fragmenting the first portion of the second data packet, fragmenting a second portion of the first available data packet into at least one data cell having a defined size.

39. The method of claim 38, wherein fragmenting the second portion of the first available data packet comprises using the at least one data element to enable fragmenting the second portion of the first available data packet.

40. The method of claim 1, wherein the user-defined number of cells comprises at least two cells.

41. The method of claim 1, wherein:

fragmenting the first portion of the first available data packet comprises using a signal processing circuit to fragment the first portion of the first available data packet; and

storing the at least one data element concerning the first available data packet comprises storing the at least one data element concerning the first available data packet in a memory included in the signal processing circuit.

42. A programmable intra-packet switching method comprising:

determining which, if any, of a plurality of data ports connected to a network contains a data packet available for processing;

fragmenting at least a first portion of a first available data packet into at least one data cell having a defined size;

monitoring the number of data cells produced to determine if a user defined number of cells have been generated;

re-determining which, if any, of the plurality of data ports contains a data packet available for processing, if it is determined that the user defined number of cells have been generated, to determine if any other port contains a data packet available for processing;

storing at least one data element concerning the data packet currently being processed if it is determined that another port contains a data packet available for processing, wherein:

the at least one data element includes a first data element indicative of the incomplete fragmentation status of said first available data packet and a second data

element selected from the group consisting of a data element indicative of the length of the portion of said data packet not fragmented and a data element indicative of the length of the portion of said data packet previously fragmented; and

the at least one data element enables fragmenting of a second portion of the first available data packet subsequent to fragmenting at least a portion of a second available data packet;

subsequent to fragmenting the first portion of the first data packet and prior to fragmenting the second portion of the first available data packet, fragmenting at least a portion of the second available data packet on a different one of the plurality of data ports; and

subsequent to fragmenting the first portion of the second data packet, fragmenting a second portion of the first available data packet.

Appendix of Evidence

None

Appendix of Related Proceedings

None